

GENERAL INSTRUCTIONS

1. Honesty Policy and Honor Code apply. Please do not cheat. **The consequence of cheating is a grade of 0.0 (fail) for the course.**
 2. Make sure that you have read and understood the problem requirements.
 3. Comply with the specifications and restrictions stated in MP Specs document, and inside the comments written in the skeleton codes. Non-compliance will result in deductions.
 4. **You are NOT allowed to use pre-defined library functions that we did not discuss in class (unless specified otherwise).**
 5. **Do NOT use float data type. Always use double data type for floating point numbers in all MP challenges.**
 6. Subject your solution to exhaustive testing to ensure that the solution is logically correct. The following scoring and deductions system will be applied.
 - A perfect score will be awarded to a compliant and logically correct solution.
 - Deductions will be applied based on the severity of the logical error. In the worst case, scenario, a score of 0 will be given (meaning the solution is logically incorrect).
 - Each unique compiler warning will result in a deduction of one point. Do not forget to use -Wall compiler option.
 - A syntax error will result in a score of 0 for the associated function (as answer to the question).
 7. Submit the required deliverables before the specified Canvas deadline.
 8. Question? Please post your question in our Canvas MP Discussion Thread. Note: I will not answer an MP related question sent via email unless it is personal in nature (for example, you and your partner would like to part ways because of disagreements in working habits).
-

The MP will be accomplished in several parts, with this document describing Part 2 activities. The activity numbers are continued from the previous part.

For Part 2, you will be writing functions that will produce answers for questions asked by another group. The questions will be assigned randomly.

Communication in any form with the group who gave the questions is STRICTLY FORBIDDEN. Violation of this directive will be considered as cheating resulting to a fail grade of 0.0.

ACTIVITY #6: Answer the questions from another group.

1. First you must represent the data values (from the SoGA dataset). The difference this time is that you are required to apply the **struct** data type. Encode in the accompanying skeleton header file **LASTNAME1_LASTNAME2.h** a **typedef** declaration for a user-defined **struct** data type whose members will be used to store ONE row of SoGA data values. This **struct** data type should NOT have any array member except for the territory name (in other words, you have to specify each risk factor as a structure member explicitly).
2. Encode in the accompanying skeleton C source file **LASTNAME1_LASTNAME2.c** the functions that will produce the answers to the 5 questions formulated by another group. The following are hard requirements that should be present in the C source file:
 - Use the **typedef alias** (that you declared in your header file) for a **1D array of struct** to represent and store **ALL** rows of the SoGA data values read from the **SoGA_DATASET.txt** file.
 - Use the **typedef alias** as well to specify all **struct** or **struct pointer** variables/pointers/parameters/function return type that you need in your function definitions.
 - At least one function definition should use a **dereference operator** (denoted by an asterisk symbol *****) **together** with a **structure member access operator** (denoted by a dot symbol **.**) to access a structure member indirectly via a structure pointer variable.
 - At least one other function definition should use a **structure pointer operator** (**->**) to access a structure member indirectly via a structure pointer variable.
3. The following requirements from MP Part 1 must also be complied with as well:
 - If the answer to a question is just one value (for example, maximum), the C function must return a value.
 - If the answer to a question involves several values (for example the names of the top 10 countries in sample question Q5), then the answer should be stored in an array or arrays that will be accessed indirectly inside the C function definition.

- The main() function should call the appropriate C function, and then call the printf() statement to display the answer to the question. **Numeric answers with double data type must be displayed with 6 digits after the decimal point.** There should NOT be any printf() and/or scanf() statement in any function definition except in main(), and in the function that reads the SoGA data text file.
 - **You must use a C double data type (NOT float) for all floating point values/variables/parameters/functions.**
 - Make sure that the answers are NOT HARDCODED inside the function definitions.
4. Submit three files, specifically (i) C source code file, (ii) header file, and (iii) a sample program output text file via the Canvas submission page for this purpose.

ACTIVITY #7: Demo your MP.

We will schedule a date/time for you to demo your project to check if it is working or not. You should know how to compile and run your program in the command line. You and your partner must both be present. The demo will be done in person (*but may be set to online mode in case of time/venue/resource related issue*).

--- End of MP Part 2 ---

Preview of MP Part 3...

MP Part 3 [20% of the MP]: expect to apply what you learned on **file processing** (covered in Chapter 4). You have the option to choose which set of questions to answer, i.e., choose either your own five questions, or those from another group that you answered in MP Part 2.